

**REMARKS**

The specification has been amended herein to change the descriptions of Figs. 9 and 16 in accordance with the substitute drawings accompanying the Notice to File Corrected Application Papers, which is being filed concurrently herewith. The specification has also been amended to add information from Figs. 9 and 16, as originally filed. No new matter has been added. As stated in section 2163.06 of the MPEP, "information contained in any one of the specification, claims or drawings of the application as filed may be added to any other part of the application without introducing new matter."

Should there be any questions regarding this paper, the Examiner is invited to contact applicants' undersigned representative at the telephone number listed below.

In the event any fees are due in connection with the filing of this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Respectfully submitted,



Himanshu S. Amin

Reg. No. 40,894

AMIN & TUROCY, LLP  
24<sup>TH</sup> Floor, National City Center  
1900 E. 9<sup>TH</sup> Street  
Cleveland, Ohio 44114  
Telephone: (216) 696-8730  
Facsimile: (216) 696-8731

**VERSION WITH MARKINGS TO SHOW CHANGES MADE**

**In the Specification:**

Please amend the specification as follows:

Page 5, lines 28-29, replace “segment of pseudocode describing a method that can be employed to identify songs” with --flowchart illustrating one method employed--.

Page 6, line 15, replace “segment of pseudocode” with --flowchart illustrating a method--.

Page 20, line 4, replace “illustrates pseudocode that” with is a flow chart that illustrates one method employed with the present invention. At 910, processing associated with the artist occurs, while at 920, processing associated with a track occurs.

Page 20, line 4, replace “an” with --the--.

Page 20, line 16, add the following:

-- For each reference track

    If artist id not same as previous reference track then

        Clear user song artist votes

        Split artist name into words, canonicalize words

        For each artist word

            Increment user song artist vote for user song artist names  
            that contain artist word

        End for

    For each user that received at least one artist vote

        Compute artist matching score

        If artist matching score is above threshold and artist matching  
            score is better than previous artist matching score, then  
                artist id of user song = artist id of reference track

        end if

    End for

---

```
End if

Select user songs whose track name matches first N characters of reference track name
If no user songs match, then return, endif

Split reference track name into words
canonicalize words
clear user song track votes
for each track word
    increment user song track vote for user song track names that contain
    track word
end for
for each user song that received at least one artist vote
    add user song artist vote to user song track vote
end for
for each user song that has at least one track vote
    compute track matching score
    if track matching score is above threshold and track matching score is
    better than previous track
        matching score user song track id = track id of reference track
    end if
end for
End for--
```

Page 27, line 1, delete "1600".

Page 27, line 1, after "is a", add flowchart illustrating a method 1600 for generating a final list. The following--.

Page 27, line 1, after "pseudocode", add can be employed to implement the method illustrated in the flowchart.

Page 27, line 2, replace “In Fig. 16” with --At 1610--.

Page 27, line 3, replace “, and” with --. At 1620,--.

Page 27, line 4, after “computed.” add the following:

--At 1630, the similarity list is generated and sorted.

    If seed song has track id then

        Seed list = seed song

    Else if seed song has artist id then

        Seed list = all user songs with same artist id

    Else

        Final list = all user songs with same artist name as seed song

        Return final list

    Endif

For each user song with track or artist id

    If user song has track id then

        User list = user song

    Else if user song has artist id then

        Check cache for similarity of seed to artist id

        If similarity value is in cache

            Add user song and similarity to final list

            Go to next user song

        End if

        User list = all user songs with same artist id

    End if

Compute limits to similarity computation

Compute similarity over all possible pairs of songs on seed list and  
user list, subject to limits

---

```
        similarity value = mean value over pairs
    if use song has no track id
        store similarity in cache
    end if
    add user song and similarity to final list
End for
```

```
Sort final list by decreasing similarity
Return final list--
```